

Using MEX Files

Phys494 Applied Fourier Analysis Lab, by İsmail Arı

September 20, 2007

This document shows how to use MEX-files in MATLAB. There are three sample problems that are solved both using MATLAB and using MEX-files.

1 Exercises

Note: For all of the MEX functions you have written, write also a MATLAB script that uses the MEX function.

1.1 Product of Two Scalars

Write a MEX function which takes two scalar inputs from the user and calculates the product.

1.2 Product of a Vector by a Scalar

Write a MEX function which takes two parameters (a scalar and a vector consequently) and calculates the elementwise product of the vector by the scalar.

1.3 Dot Product of Two Vectors

Write a code to calculate the dot-product of two input vectors, $y = \vec{a} \cdot \vec{b}$. Your program will take \vec{a} and \vec{b} as inputs. You need to check whether the length of the vectors are the same or not, if not warn the user. You may also check whether the inputs are vectors or not.

1.4 Product of Two Complex Scalars

Write a MEX function which takes two scalar inputs from the user and calculates the product but the inputs are complex instead of real values.

1.5 Dot Product of Two Complex Vectors

Write a code to calculate the dot-product of complex vectors, $y = \vec{a} \cdot \vec{b}$. Your program will take two inputs and return only one output. You can assume the inputs are vectors, you don't need to check inputs, assume the user gives correct inputs.

2 Solutions

Solutions are attached to this file. The MATLAB file involves the code that calls the MEX functions in it. Open the files and go through the code. The comments are supposed to be satisfactory.

To compile the MEX-files into dll's, try the following step by step:

- Set the current folder as working folder of MATLAB **OR** Copy these example files to "<MATLAB INSTALLED FOLDER>/Work" folder
- At MATLAB command prompt, type

```
mex product1_MEX.c
```

(Note: If you are doing this for the first time, MATLAB will ask you some options for the compilation processes. You can select MATLAB's own compiler.)

- **mex** function creates a dll for the MEX file with the same name. (A *DLL - Dynamic Link Library*- is a file that can be loaded and executed by programs dynamically. Basically it's an external code repository for programs. Since usually several different programs reuse the same DLL instead of having that code in their own file, this dramatically reduces required storage space. A synonym for a DLL would be library.)

(Note: Newer versions of MATLAB creates mex32 or mex64 extensions instead of dll but they work in the same manner.

- This means that when you call your MEX routine in MATLAB, it will call this dll file.
- Create dll's for all MEX functions and then run the attached MATLAB code that uses the MEX routines and see the results.

For further information about mex-files, you can refer to:

- <http://www.mathworks.com/support/tech-notes/1600/1605.html> *MEX-files Guide @ MATHWORKS*
- <http://www.codeproject.com/samples/mexFunction.asp> *Sample Code and a Small Tutorial*
- http://prism.mem.drexel.edu/Shah/public_html/mex2c.html *Tutorial for Creating MEX files*

3 Exercise

Write a code to calculate the cross-product of vectors $\vec{x}_1 = (a_1 \ a_2 \ a_3)$ and $\vec{x}_2 = (b_1 \ b_2 \ b_3)$, i.e. $\vec{y} = \vec{x}_1 \times \vec{x}_2$. Your program will take two inputs of length three and return only one output of length three.